

IN THE CLAIMS

BELOW IS A COMPILATION OF ALL CLAIMS DISCUSSED IN THIS RESPONSE, BOTH AMENDED AND UNAMENDED.

Please cancel claims 39-46, without prejudice, the subject matter of those claims to be reserved for prosecution in a separate application.

1. (PREVIOUSLY AMENDED) A computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory, and nonvolatile storage, the computerized game controller being operable to control a computerized wagering game;

an operating system comprising: a system handler application operable to dynamically link with at least one gaming program object;

an Application Program Interface having functions callable from the gaming program object and

an operating system kernel that executes the system handler application.

2. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises an event handler.

3. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises software having the ability when executed to:

unload a previous gaming program object if a previous object has been loaded;

load a new gaming program object; and

execute the new gaming program object.

4. (PREVIOUSLY AMENDED) The computerized wagering game apparatus of claim 1, wherein data variables modified by the gaming program objects are stored in nonvolatile storage, and functions verify that the operating system or code for a shared object has not changed.
5. (ORIGINAL) The computerized wagering game apparatus of claim 4, further comprising a game state device providing a variable name index to associated variable data locations within the nonvolatile storage.
6. (PREVIOUSLY AMENDED) The computerized wagering game apparatus of claim 4, wherein changing a data variable in nonvolatile storage causes execution of a corresponding callback function in the system handler application gaming program object.
7. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the computerized game controller comprises an IBM PC-compatible computer.
8. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the operating system kernel is a Linux operating system kernel.
9. (ORIGINAL) The computerized wagering game apparatus of claim 8, wherein the Linux operating system kernel has at least one selected device handler disabled.
10. (ORIGINAL) The computerized wagering game apparatus of claim 9, wherein the at least one selected device handler that is disabled is selected from the group consisting of a keyboard handler, an I/O port handler, a network interface handler, a storage device controller handler, and a I/O device handler.

11. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises an API with functions callable from the gaming program objects.

12. (ORIGINAL) The computerized wagering game apparatus of claim 11, wherein the API comprises functions that are specific to a computerized gaming apparatus.

13. (PREVIOUSLY AMENDED) A method of managing data in a computerized wagering game apparatus via a system handler application, comprising:

loading a first program object and providing an Application Program Interface having functions called by the first program object,

executing the first program object,

storing data variables in nonvolatile storage, such that a second program object later

loaded can access the data variables in nonvolatile storage,

unloading the first program object, and

loading a second program object.

14. (ORIGINAL) The method of claim 13, further comprising executing a corresponding callback function upon alteration of variable data in nonvolatile storage.

15. (ORIGINAL) The method of claim 13, further comprising handling events via the system handler application.

16. (PREVIOUSLY AMENDED) A secure computerized wagering game system controlled by a general-purpose computer, comprising an operating system kernel that is customized to disable selected device handlers, and a program that loads a first program game object and

the first program game object calls up a function from within an Application Program Interface.

17. (PREVIOUSLY AMENDED) A secure computerized wagering game system controlled by a general-purpose computer comprising nonvolatile storage that stores program variables, such that loss of power does not result in loss of the state of the computerized wagering game system, and a program that loads a first program game - object and the first program game object calls up a function from within an Application Program Interface.

18. (ORIGINAL) The secure computerized wagering game system of claim 17, further comprising at least one gaming program object, such that a single gaming program object is loaded and executed at any one time but gaming program objects are operable to share data via the program variables in nonvolatile storage.

19. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the medium being within a wagering apparatus, the instructions when executed operable to cause a computer to:

cause a system handler application to load and execute gaming program objects;

cause a loaded gaming program object to call up a library of functions provided by the system handler application;

load a first program object from the library,

execute the first program object, including having the first program object call up a function from within an Application Program Interface,

store data variables in nonvolatile storage, such that a second program object in the library later loaded can access the data variables in nonvolatile storage,

unload the first program object, and

load the second program object, including having the second program object call up a function from within an Application Program Interface.

20. (ORIGINAL) The machine-readable medium of claim 19, with further instructions operable when executed to cause a computer to execute a corresponding callback function upon alteration of variable data in the nonvolatile storage.

21. (ORIGINAL) The machine-readable medium of claim 19, with further instructions operable when executed to cause a computer to manage events via the system handler application.

22. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the instructions when executed operable to cause a computer to manage at least one gaming program object via a system handler application, the gaming program object calling up a function from within an Application Program Interface such that a single gaming program object is loaded and executed at any one time but gaming program objects are operable to share data via the program variables in a nonvolatile storage.

23. (ORIGINAL) The machine-readable medium of claim 22, with further instructions operable when executed to cause a computer to provide functions through an API that comprises a part of the system handler application.

24. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the instructions when executed operable to cause a computer to manage at least one gaming program object via a system handler application, such that a single gaming program object is executed at any one time, wherein gaming program objects are operable to share data in nonvolatile storage and to call up a function from within an Application Program Interface.

25. (ORIGINAL) The machine-readable medium of claim 21, wherein only one gaming program object executes at any one time.
26. (PREVIOUSLY AMENDED) The machine-readable medium of claim 21, with further instructions operable when executed to cause a computer to provide functions through the API that comprises part of the system handler application.
27. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the instructions when executed are operable to store game data in non-volatile storage, such that the state of the computerized wagering game system is maintained when the machine loses power, and wherein gaming program objects are operable to share data in the nonvolatile storage and to call up a function from within an Application Program Interface.
28. (PREVIOUSLY AMENDED) A gaming machine architecture, comprising an operating system, and a plurality of shared objects; wherein each shared object describes game personality in a selected mode, and wherein gaming program objects are operable to share data in nonvolatile storage and to call up a function from within an Application Program Interface.
29. (ORIGINAL) The gaming machine architecture of claim 28, wherein the operating system comprises an IBM PC-based operating system.
30. (ORIGINAL) The gaming machine architecture of claim 28, wherein the operating system comprises a system handler.
31. (ORIGINAL) The gaming machine architecture of claim 30, wherein the system handler comprises a plurality of device handlers.
32. (ORIGINAL) The gaming machine architecture of claim 30, wherein the system handler comprises an event queue.

33. (PREVIOUSLY AMENDED) The gaming machine architecture of claim 30, wherein the system handler comprises a plurality of API callable functions, callable by the shared objects.
34. (ORIGINAL) The apparatus of claim 1, wherein the system handler comprises an event queue that determines the order of execution of each specified device handler.
35. (ORIGINAL) The apparatus of claim 1, wherein the system handler comprises an API having a library of functions.
36. (ORIGINAL) The apparatus of claim 34, wherein the event queue is capable of queuing on a first come, first serve basis.
37. (ORIGINAL) The apparatus of claim 34, wherein the event queue is capable of queuing using more than one criteria.
38. (ORIGINAL) The apparatus of claim 1, wherein the system handler and kernel work in communication to hash the system handler code and the operating system kernel code.
39. (CANCELLED)
40. (CANCELLED)
41. (CANCELLED)
42. (CANCELLED)
43. (CANCELLED)
44. (CANCELLED)
45. (CANCELLED)

46. (CANCELLED)

47. (PREVIOUSLY AMENDED) A method of modifying an operating system kernel in a gaming apparatus where a program object calls up a function from within an Application Program Interface, comprising at least one modification to obtain functionality selected from the group consisting of:

- 1) accessing user level code from ROM;
- 2) executing user level code from ROM;
- 3) zeroing out unused RAM;
- 4) testing and/or hashing the kernel to verify that the operating system kernel or a code for a shared object has not changed;
- 5) disabling selected device handlers.

48. (ORIGINAL) The computerized wagering game of claim 1 wherein the operating system kernel executing on the computerized game controller comprises an element of a universal operating system also comprising a system handler.

49. (ORIGINAL) The computerized wagering game of claim 1 wherein the non-volatile storage is controlled by a general-purpose computer, the non-volatile storage stores game data, and the storage of game data in the non-volatile storage preserves the state of the computerized wagering game system upon loss of power.

50. (ORIGINAL) The computerized wagering system of claim 1 operating within a networked on-line system.

51. (ORIGINAL) The computerized wagering system of claim 1 wherein the system controls a progressive meter.

52. (PREVIOUSLY AMENDED) The computerized wagering system of claim 1 wherein an Application Program Interface is also dynamically linked from the systems handler.

53. (PREVIOUSLY AMENDED) The machine readable medium of claim 19 wherein the instructions when executed are operable to dynamically link an Application Program Interface to a gaming program object.

54. (PREVIOUSLY AMENDED) The machine readable medium of claim 22 wherein the instructions when executed are operable to dynamically link an Application Program Interface to a gaming program object.

55. (PREVIOUSLY ADDED) The computerized wagering game apparatus of claim 1 wherein the operating system operates a function to verify that the operating system kernel or a code for a shared object has not changed.

56. (PREVIOUSLY ADDED) A method of managing data in a computerized wagering game apparatus via a system handler application according to claim 1 wherein a function is performed to verify that an operating system kernel or a code for a shared object has not changed.

57. (PREVIOUSLY AMENDED) The computerized wagering game apparatus of claim 1 with instructions thereon wherein a function is performed to verify that an operating system kernel or a code for a shared object has not changed